

INF-253 Lenguajes de Programación

Profesores: José Luis Martí Lara, Wladimir Ormazabal Orellana

Ayudante coordinador: Bastián Ríos Lastarria

Ayudantes: Fernanda Araya, Diego Duarte, Andrés Jablonca, Cristian Tapia

March 21, 2025

1 Caos contraseñico!

¡Felicidades! Eres el nuevo Administrador Oficial de Contraseñas de Soluciones Sansaniles, una empresa donde las reglas para contraseñas cambian más rápido que el clima. El jefe, famoso por sus cambios de ánimo y su odio por las vocales los martes, te ha encomendado una misión crítica: "Crea un sistema que gestione contraseñas bajo mis diversas reglas diarias, permita fusionarlas como si fueran cocktails, y asegúrate de que nadie use el número 7 los viernes. ¡Confío en ti!" Su tarea es desarrollar un programa en Python que:

- Valide, cree y modifique contraseñas según reglas activas.
- Gestione usuarios (altas, bajas, cambios).
- Implemente las operaciones sorpresa que tiene preparadas el jefe.
- Utilice solo Python 3 y expresiones regulares (regex).

2 Especificaciones del Sistema Sansanil

El jefe te dará un archivo `'logSansano.txt'` donde estarán todas las instrucciones, reglas y empleados con sus contraseñas nuevas del día, el programa deberá poder **leer las instrucciones y colocar las restricciones descritas por el jefe en el "logSansano.txt"** para que aquellas contraseñas que no están de acuerdo al ánimo del jefe no pasen al sistema.

Deberá **gestionar** los nuevos empleados y sus intentos de colocar una contraseña en su cuenta, es decir, **crear nuevos usuarios y sus respectivas contraseñas, editarlas y borrarlas**, y principalmente **las reglas deben ser a prueba de cambios**.

Ya que el jefe es famoso por sus cambios de humor, es muy posible que a mitad del día se le ocurra quitar o modificar alguna regla; **aquellos que ya registraron su contraseña estarán a salvo, pero los que no llegaron a tiempo no corren con la misma suerte**, ellos deberán seguir las nuevas reglas del nuevo estado anímico del jefe.

Cabe detallar que el jefe también esperará un archivo con todos los cambios realizados por el sistema para asegurarse que hayas realizado un trabajo bien hecho, así que ¡más te vale automatizar eso también!

2.1 Reglas Dinámicas: El Capricho del Jefe

Aquí se detallará todas las posibles instrucciones que puede dar el jefe y lo que se espera que suceda con cada una de ellas. En los ejemplos habrán más casos para mayor claridad (las variables [ya sean numéricas o letras] serán representadas con un x, y o z minúscula).

- **SIZE RULE x** /se espera que la contraseña tenga un largo "x" o más. (solo numérico)
- **NEW RULE NUMEROS x,y** /se agrega la regla "NUMEROS" que fuerza que las siguientes contraseñas no contengan ningún "x" ni "y".
- **ADD_TO RULE NUMEROS z** /se le agregará a esa regla el número mencionado para que NO acepte ninguna contraseña que contenga "y" o "z".
- **REMOVE_FROM RULE NUMEROS y** /se espera que ahora el "y" sí se pueda incluir en las contraseñas.
- **DELETE RULE NUMEROS** / se espera que la regla "NUMEROS" sea eliminada (recordar "NUMEROS" es solo un nombre, en el EBNF está aclarado a mayor detalle este tipo de cosas).
- **NEW PASSWORD FROM z contraseña** / considerando "z" un usuario nuevo, se le asigna una contraseña nueva. (en este caso, literalmente "contraseña")
- **CHANGE PASSWORD FROM z contraseña_antigua contraseña_nueva** /se espera que la contraseña que tenía "z" sea reemplazada por una nueva.
- **SHOW PASSWORD FROM z** /mostrar la contraseña actual del usuario "z"
- **DELETE PASSWORD FROM z** /se espera que el usuario "z" quede sin contraseña en su cuenta.
- **FUSION x y INTO z** /este comando especial se espera que combine dos contraseñas existentes de los usuarios "x" e "y" y la coloque como contraseña del usuario "z".

2.2 Aspecto del log Sansano

El aspecto general que tendrá **logSansano.txt** se ejemplifica a continuación:

- **logSansano.txt**

```
SIZE RULE 9
NEW PASSWORD FROM User1 holAa123
NEW RULE Impares 1,3,5,7,9
NEW PASSWORD FROM usuario2 HolaMundo42
ADD_TO RULE Ipares 0,2,4,6,8
NEW RULE Pares 0,2,4,6,8
NEW PASSWORD FROM Usuario105 contraSinNumeros
NEW PASSWORD FROM EmpleadoAtrasado LlegueTarde123
DELETE RULE Impares
```

```
DELETE RULE Pares
SHOW PASSWORD FROM Usuario105
FUSION usuario2 Usuario105 INTO Usuario105
DELETE PASSWORD FROM usuario2
SHOW PASSWORD FROM usuario2
```

2.2.1 Explicacion:

Del siguiente ejemplo se puede extraer lo siguiente:

1. Primero, solo serán aceptadas las contraseñas de largo 9 o más, "holAa123" al ser de 8 caracteres, quedaría fuera y no debe ser aceptada; más abajo se explica qué hacer en casos de rechazo y aceptación.
2. La regla "Impares" prohíbe el uso de numeros impares y los elementos están separados por ","
3. Al haber un error ortográfico en el comando "ADD TO", esa regla será rechazada por el sistema.
4. Las contraseñas de usuario2 y Usuario105 no deben tener problemas ya que están según las reglas escritas, no así la de EmpleadoAtrasado, el cual contiene números por lo que debe ser rechazado.
5. Lo que hace FUSION es mezclar las contraseñas "HolaMundo42" y "contraSinNumeros", para que la nueva contraseña de Usuario105 sea "HolaMundo42contraSinNumeros"
6. Al eliminar la contraseña de usuario2 y luego pedir que se muestre, el sistema debe reconocer que ya no existe una contraseña para usuario2 y mostrar un mensaje de error.

2.3 Resultados.txt

Luego de cumplir con los caprichos del jefe y crear tu programa a prueba de cambios anímicos, todos los cambios hechos ya sean confirmados o rechazados necesitarán de un registro donde el jefe pueda verificar que esté todo en orden. Queda en manos del programador la forma en la que se manejen los errores y contraseñas válidas pero un ejemplo de registro final correcto debe verse así:

• Resultados.txt

Nueva regla de tamaño de contraseña impuesta con éxito.
Error en crear contraseña para User1, tamaño insuficiente.
Nueva regla "Impares" creada con éxito.
Contraseña para usuario2 establecida con éxito.
Error al añadir elementos a "Ipares", regla no encontrada.
Nueva regla "Pares" creada con éxito.
Contraseña para Usuario105 establecida con éxito.
Error en crear contraseña para EmpleadoAtrasado, no cumple con regla "Impares".
Regla "Impares" eliminada con éxito.
Regla "Pares" eliminada con éxito.

contraSinNumeros

Fusión realizada con éxito.

Contraseña de usuario2 eliminada con éxito.

Contraseña para "usuario2" no encontrada. ¿Está escribiendo el nombre correctamente?

Recordar, esto es solo un ejemplo, lo importante es notificar los cambios, el cómo se realiza es cuestión de cada programador/a.

2.4 EBNF General

digito ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

digito_o_cero ::= digito | '0'

entero ::= (digito {digito_o_cero}) | '0'

string ::= (A-Z | a-z) {A-Z | a-z | entero}

variable ::= ([string] | [entero])

2.4.1 REGLAS

regla ::= 'RULE'

regla_tamaño ::= 'SIZE' [regla] [entero]

nueva_regla ::= 'NEW' [manejo_de_reglas] {'', 'variable'}

sumar_a_regla ::= 'ADD_TO' [manejo_de_reglas] {'', 'variable'}

quitar_de_regla ::= 'REMOVE_FROM' [manejo_de_reglas] {'', 'variable'}

eliminar_regla ::= 'DELETE' [regla] [string]

manejo_de_reglas ::= [regla] [string] [variable]

2.4.2 CONTRASEÑAS

contraseña ::= 'PASSWORD'

nueva_contraseña ::= 'NEW' [sentencia_contraseña] ([string]||[entero])

editar_contraseña ::= 'CHANGE' [sentencia_contraseña] ([string]||[entero]) ([string]||[entero])

eliminar_contraseña ::= 'DELETE' [sentencia_contraseña]

mostrar_contraseña ::= 'SHOW'

sentencia_contraseña ::= [contraseña] 'FROM' [string]

2.4.3 ESPECIAL

fusion ::= 'FUSION'

realizar_fusion ::= [fusion] [string] [string] 'INTO' [string]

3 Validador.py

La tarea a entregar será un código en Python llamado "**Validador.py**" el cual será el programa a petición del jefe. Este código será el encargado de validar, crear, modificar, y eliminar contraseñas según las reglas que serán entregadas junto a las contraseñas en el archivo "**logSansano.txt**" según lo explicado en las secciones anteriores (2.1, 2.2).

Además, deberá generar un archivo de texto llamado "**Resultados.txt**" que cumpla con lo mencionado en la subsección 2.3.

A continuación, un ejemplo con algunos casos borde mas el registro de salida esperado para una mayor claridad de lo que hay que realizar:

- logSansano.txt

```
NEW PASSWORD FROM UsuarioRandom Contr4sinRestr1cci0n3s
SIZE RULE 9,2
NEW RULE regla1 0,1,3,4
ADD_TO RULE regla1 a,b,c,d,e,f,g,h
CHANGE PASSWORD FROM UsuarioRandom Contr4sinRestr1cci0n3s contraNueva123
REMOVE_FROM RULE regla1 1,2,3,a,c,e,x,y,z
CHANGE PASSWORD FROM UsuarioRandom Contr4sinRestr1cci0n3s contraNueva123
NEW PASSWORD FROM UsuarioEspecial !@#$!?!?!?
DELETE RULE regla1
DELETE RULE regla1
NEW RULE regla1 A,B,C,D,E
NEW PASSWORD contrasinmayus
change password contrasinmayus CONTRACONMAYUS
```

- Resultados.txt

```
Contraseña para UsuarioRandom establecida con éxito.
Error en formato al intentar crear regla de tamaño.
Nueva regla "regla1" creada con éxito.
Elementos añadidos a "regla1" con éxito.
Error al intentar cambiar contraseña de UsuarioRandom, nueva contraseña no permitida.
Elementos eliminados de regla1 con éxito.
Contraseña para UsuarioRandom cambiada con éxito.
Error al crear contraseña para UsuarioEspecial, caracteres no permitidos.
Regla "regla1" eliminada con éxito
```

Error al intentar eliminar regla1, regla no existente.

Nueva regla "regla1" creada con éxito.

Error de sintaxis al intentar crear una contraseña. Elementos faltantes

Error de sintaxis. Comando no reconocido.

4 Sobre la entrega

- El código debe venir indentado y ordenado.
- Se debe entregar los siguientes archivos:
 - Validador.py
 - README.txt
- Si no existe orden en el código habrá descuento.
- **Al no hacer uso de expresiones regulares, la tarea no será revisada.**
- Solo se permite el uso de la biblioteca estándar de Python (docs.python.org/library) (Incluye RegEx). No se permite usar otros módulos externos.
- Todas las funciones deben ir comentadas con el siguiente formato:

```
"""
***
Parametro 1 : Tipo
Parametro 2 : Tipo
...
***
Tipo de Retorno o None
***
Breve descripción de la función y el retorno
"""
```

Se harán descuentos por funcion no comentada

- La entrega debe realizarse en un archivo comprimido en tar.gz y debe llevar el nombre: Tarea1LP_RolAlumno.tar.gz.
Ej.: Tarea1LP_202373000-k.tar.gz.
- El archivo README.txt debe contener el nombre y rol del alumno e instrucciones detalladas para la correcta utilización de su programa.
- La entrega será vía aula y el plazo máximo de entrega es hasta el día **miércoles 02 de abril, 23:55 horas, vía aula.**
- Por cada día de atraso se descontarán 20 puntos (10 puntos dentro de la primera hora)
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.
- Solo se pueden realizar consultas respecto a la tarea hasta 2 días antes de la entrega.
- Las consultas del enunciado se debe realizarse mediante el foro de la tarea disponible en AULA.

5 Calificación

5.1 Entrega

Para la calificación de su tarea, debe realizar una entrega con requerimientos mínimos que otorgarán 30 pts base, luego se entregará puntaje dependiendo de los otros requerimientos que llegue a cumplir.

5.1.1 Entrega Mínima (total 30 pts)

- Crear expresiones regulares que detecten y validen la **sintaxis** de los comandos **NEW RULE** y **NEW PASSWORD**, extrayendo sus parámetros para su procesamiento. (10 pts)
- Procesar los comandos **NEW RULE** y **NEW PASSWORD**, validando contraseñas según las reglas activas y **genera un archivo de salida**. (20 pts)

Ejemplo caso de entrega mínima:

- logSansano.txt

```
NEW RULE NONUMBERS 1,2,3,4,5
NEW PASSWORD FROM Usuario1 SnsaN0
NEW PASSWORD FROM Usuario2 123abc
```

Archivo de salida:

- Resultados.txt

Nueva regla "NONUMBERS" creada con éxito.
Contraseña para Usuario1 establecida con éxito.
Error al crear contraseña para Usuario2, caracteres no permitidos.

De obtener los 30 puntos de la entrega mínima, se procederá a la revisión de las siguientes partes de la tarea.

5.1.2 Asignación de puntajes por implementación de comandos (total 40 pts)

- Correcta implementación y funcionamiento de 'SIZE RULE'. (8 pts)
- Correcta implementación y funcionamiento del comando 'DELETE' para reglas y contraseñas. (8 pts)
- Correcta implementación y funcionamiento de 'NEW RULE...' y 'NEW PASSWORD...' en casos avanzados y complejos. (5 pts)
- Correcta implementación y funcionamiento de 'REMOVE_FROM'. (3 pts)
- Correcta implementación y funcionamiento de 'ADD_TO'. (3 pts)
- Correcta implementación y funcionamiento de 'CHANGE PASSWORD...'. (4 pts)
- Correcta implementación y funcionamiento de 'SHOW PASSWORD...'. (4 pts)
- Correcta implementación y funcionamiento del comando 'FUSION'. (5 pts)

5.1.3 Asignación de puntajes por detección y notificación de errores (total 30 pts)

- El programa detecta correctamente todos errores de sintaxis y se informa a través del registro de salida. (10 pts)
- Se registra correctamente todas las reglas establecidas en el archivo de salida y se detecta e informa correctamente errores relacionados al mal uso de los comandos "RULE". (10 pts)
- El programa es capaz de validar correctamente y notificar todos los errores que puedan existir al ingresar nuevas contraseñas, cambiarlas o fusionarlas. (10 pts)

5.2 Descuentos

- Falta de comentarios (-10 pts c/u MAX 30 pts)
- Falta de README (-20 pts)
- Falta de alguna información obligatoria en el README (-5 pts c/u)
- Falta de orden (-20 pts)
- Día de atraso (-20 pts por día, -10 dentro de la primera hora)
- Mal nombre en algún archivo entregado (-5 pts c/u)

En caso de existir nota negativa, esta será remplazada por un 0.